CS-168 Project Proposal: Extending Bitcoin Features to Spartan Gold

Team Members

Anant Joshi
Velonjatovo Andrilalaina Obeda

Introduction

Our goal for the project is to extend some of the features of Bitcoin into Spartan Gold. Mainly, we will be focusing on two main features: Merkle Trees and Proof-of-work target adjustments based on the power of the network. These features and some of our ideas on how to accomplish these tasks are described in details in the following sections. Our plans to implement these features are briefly described in the **Objectives** section.

Merkle tree

The Merkle Tree is the a data structure used in cryptography to ensure the efficiency and the security of data in the tree. The tree is structured in way that the leaves of the tree represent individual transactions and the non-leaf nodes are the hashes of their two child nodes. The process is repeated recursively until we are left with a single node, which becomes the Merkle Root. The main benefits of the this data structure is that it facilitates the verification of a given hash when checking whether or not the hash is included in the tree. Additionally, the Merkle Tree is also scalable if the amount of data increases without losing the efficiency of verification.

Set the proof of work difficulty to adjust to the power of network over time

The proof of work mechanism consists of miners guessing the nonce that is used to compute a hash smaller than the target hash. In Bitcoin blockchain, the average rate of miners generating one Bitcoin block is 10 minutes. Hence, if the computational power or the number of miners increases and a block is being generated in less than 10 minutes, the proof of work difficulty needs to increases as well. The inverse aslo applies, if the average hash rate of the network mining takes longer than 10 minutes, the difficulty is decreased.

To adjust the proof of work difficulty, we can change the target value. To increase the level of difficulty, we make the target value so that it would be harder for miners to find hash value lower that the target value. The opposite is also true.

Objectives

To implement and two feature to the Spartan Gold -

1. Add a Merkle Tree data structure to Spartan Gold
    - Stretch goal to implement efficiency improvements to Merkle Trees. (Currently, looking at implementing Fast Merkle Trees if possible)
2. Set the proof of work difficulty to adjust to the power of the network over time
    - Regulate the proof of work difficulty based of the average time that miners take to find a proof
    - Preventing centralization by keeping the level of difficulty hard enough so that mining pools or individuals with high computational power won’t be able to dominate the network

Plan

Week	Goal
Week 1 (Apr 10 - Apr 16)	Start working on both Merkle Tree and PoW Target Improvements
Week 2 (Apr 17 - Apr 23)	Merkle Tree Implementation
Week 3 (Apr 24 - Apr 30)	PoW Target Change Implementation
Week 4 (May 1 - May 6)	PoW Target Change Implementation
Week 5 (May 7 - May 13)	Debug/Fix and Finalize Implementation

Expected Outcomes

Working implementation of both Merkle Trees and Adjusting PoW (proof-of-work) Target in Spartan Gold.